

A Coruña, Septiembre 2020



VALICO

THE GRIM WORLD

Trabajo de Fin de Grado por

Iván Fernández López
Luis David Lobo Pérez



Tutorizado por

Antonio Seoane Nolasco



Índice

1.	Introducción.....	4
2.	Descripción del proyecto.....	5
	2.1 El videojuego.....	6
	2.2 Objetivos del proyecto.....	7
	2.3 Público objetivo.....	8
	2.4 Pipeline.....	9
	2.5 Medios técnicos.....	11
3.	Preproducción.....	12
	3.1 Conceptualización.....	13
	3.2 Ambientación.....	14
	3.3 Mecánicas planteadas.....	17
	3.4 Creación de la demo.....	18
4.	Producción.....	19
	4.1 Creación de escenarios.....	20
	4.2 Creación de personajes.....	24
	4.3 Creación de props.....	35
	4.4 Diseño sonoro.....	37
	4.5 Interfaz.....	40
	4.6 Jugabilidad y mecánicas.....	41
	4.7 Cinemáticas.....	48
5.	Postproducción.....	51
6.	Testing.....	53
7.	Plan de producción.....	55
	7.1 Calendario de hitos.....	56
	7.2 Presupuesto.....	57

8.	Conclusiones.....	59
9.	Bibliografía.....	62

Anexos

1.	Documento de diseño
2.	Listado de Eventos
3.	Listado de misiones
4.	Guía de Valico: The Grim World
5.	Lore de Valico
6.	Gameplay



1. Introducción



1. Introducción

Este Trabajo de Fin de Grado es el desarrollo del prototipo del videojuego Valico: The Grim World, que fue diseñado en la asignatura de Videojuegos y posteriormente adaptado para el desarrollo de este proyecto.

El reto principal del TFG fue el desarrollo de la narrativa de la demo, adaptando la historia del juego completo, para centrarse más en la parte mecánica y estética.

Este prototipo es el resultado de recolectar las ideas planteadas en el documento de diseño del videojuego. (Anexo 1)



2. Descripción del proyecto



2.1 El videojuego

Valico: The Grim World es un videojuego en tercera persona de acción y aventura en tres dimensiones, inspirado en la cultura céltico-galaica.

Valico es el mundo donde se ambienta el videojuego, una tierra fantástica de mitología y cultura celta. En ella vive Cassidy, el último miembro de una orden de Meigas que un día desaparece misteriosamente. Motivado por la enfermedad de su mejor amigo, Gallagher y la ayuda de la diosa Morrigan, Cassidy decide emprender un viaje para terminar con la enfermedad que asola su tierra.

A lo largo de su aventura, Cassidy encontrará amistades, peligros y retos que deberá superar con su fuerza y astucia, para finalmente salvar Valico de la oscuridad y las garras de Bandua.

Para más información consultar el documento de diseño del videojuego. (Anexo 1)



Icono de Valico: The Grim World

2.2 Objetivos del proyecto

El objetivo del TFG a corto plazo es conseguir el desarrollo de una demo jugable que contenga todo lo necesario a nivel mecánico para visualizar una idea global y condensada de lo que se define en el documento de diseño. (Anexo 1)

También, a largo plazo se quiere desarrollar el videojuego entero, con los conocimientos adquiridos y las pruebas realizadas en este proyecto, para posteriormente comercializar Valico: The Grim World, que busca hacerse un hueco en el mercado con una temática celta poco explotada en los videojuegos y siguiendo la estela de los juegos de The Legend of Zelda tridimensionales.

Cuando Valico: The Grim World esté completo, se quiere lograr la difusión de la mitología celta, así como fomentar una imagen más cercana de la cultura y tradiciones gallegas.

También, se quiere aportar a la industria del videojuego creando un producto innovador a nivel mecánico. Valico tiene como objetivo dar una

una sensación de libertad y espectacularidad al combatir con magia elemental y armas clásicas cuerpo a cuerpo.



The Legend of Zelda: Ocarina of Time. Nintendo, 1998

2.3 Público objetivo

El proyecto está enfocado para un público habituado a juegos de aventura, acción y, también, a juegos de rol. El título está pensado principalmente para el disfrute de un público casual, sin embargo, Valico ofrece distintos retos y características muy atractivas para jugadores más hardcore.

El perfil del jugador de Valico es de una persona de más de 20 años, con un leve nivel de cultura en las sociedades antiguas; con bagaje en las sagas de The Legend of Zelda, Dragon Quest, Jack and Daxter, Ratchet & Clank o Diablo; es un jugador ya criado, como mínimo, durante la quinta generación de consolas y ha pasado por las diferentes etapas de la historia del videojuego.



Dragon Quest VIII: Journey of the Cursed King. Square Enix, 2004.

2.4 Pipeline

El desarrollo del prototipo de Valico: The Grim World siguió un pipeline, que a medida que la producción avanzó fue cambiando hasta conseguir la forma más eficaz de trabajar.

Uno de los factores más importantes en los cambios del pipeline fue la pandemia mundial vivida este año, sin embargo, el equipo buscó la forma de sacar adelante el trabajo de la mejor manera posible.

El origen del proyecto es el documento de diseño del videojuego, en donde se describen los personajes, mecánicas, narrativa, etc. Posteriormente, se plantearon los niveles a desarrollar.

El primer paso del desarrollo fue el blocking de los niveles, al mismo tiempo se comenzó el proceso de creación y obtención de props y se inició la creación del personaje jugable, Cassidy.

Luego se comenzó la construcción de los niveles con las piezas definitivas, a la par que se programaban las mecánicas de Cassidy.

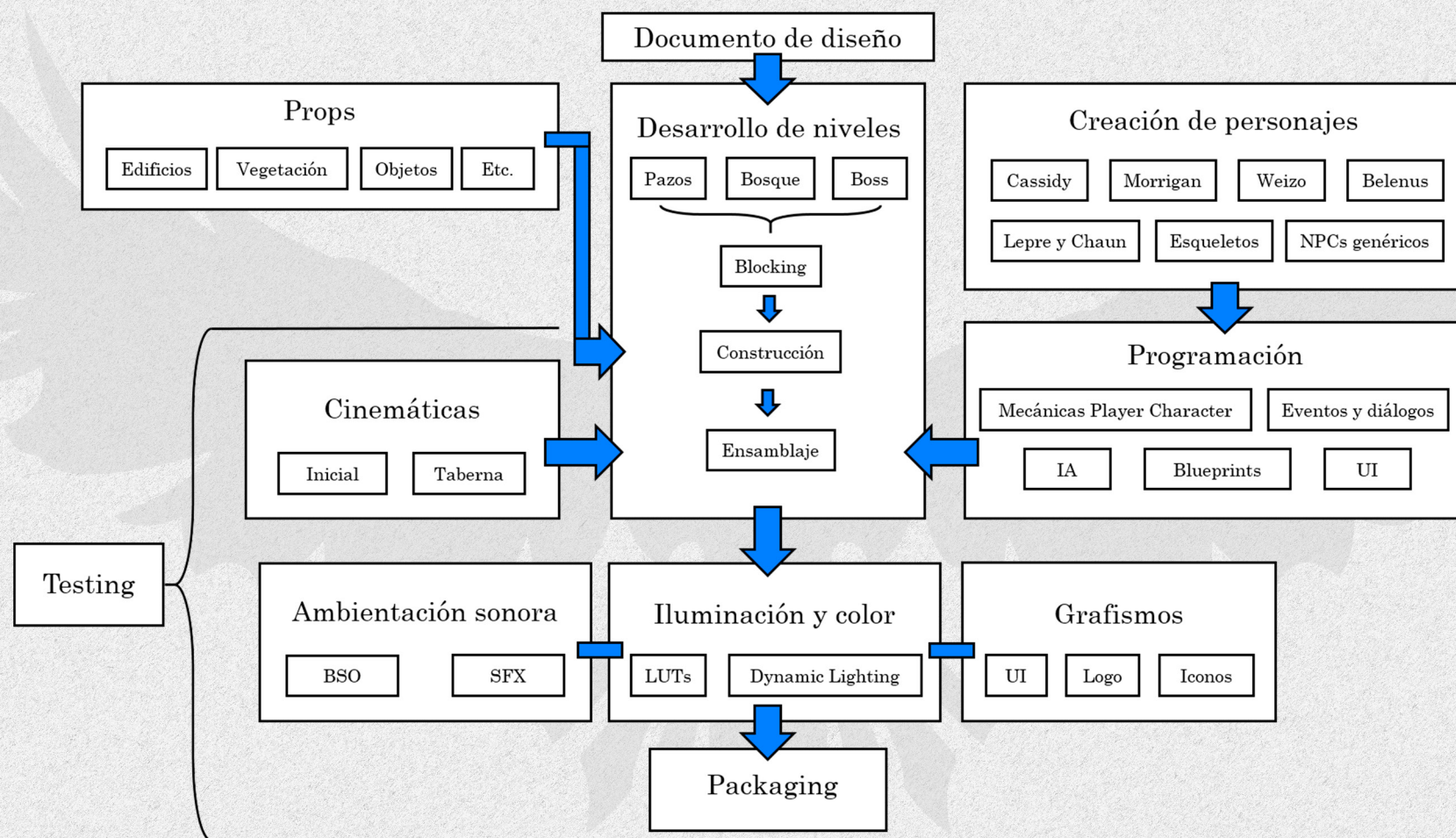
Con la estructura del nivel lista y las acciones esenciales del protagonista funcionando, se pasó a la creación del resto de personajes.

Una vez finalizado el proceso anterior, se pasó a la programación de la IA, eventos, blueprints, cinemáticas y widgets necesarios para el funcionamiento del nivel. Al mismo tiempo, personas externas al proyecto hicieron pruebas para detectar fallos, que permitieron al equipo hacer las correcciones pertinentes.

Luego se hizo la ambientación sonora del juego y los ajustes de luz y color. También se crearon y agregaron los grafismos finales para la interfaz del juego.

Con la demo terminada, se hicieron las pruebas finales, se afinaron los últimos detalles y se procedió a empaquetar el juego.

En el siguiente esquema se puede visualizar mejor el pipeline seguido.



2.5 Medios técnicos

Los programas utilizados para el desarrollo del proyecto fueron:

Motor de videojuegos

Unreal Engine 4

Modelado, animación y texturizado 3D

Autodesk Maya, Blender, Marvelous Designer 9, VRoidStudio, Fuse, Mixamo, Substance Painter

Diseño gráfico

Adobe: Photoshop, InDesign

Montaje y edición de audio y vídeo

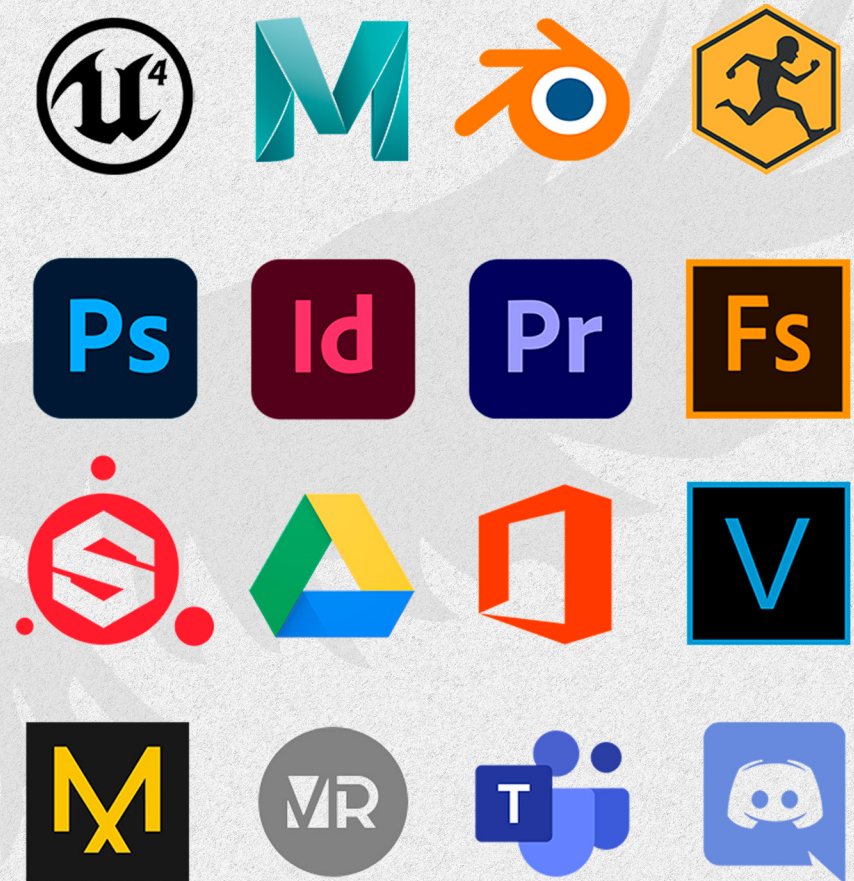
Adobe Premiere Pro, Sony Vegas Pro 15

Gestión de documentos

Google Drive, Microsoft Office: Word, Excel, PowerPoint

Comunicación

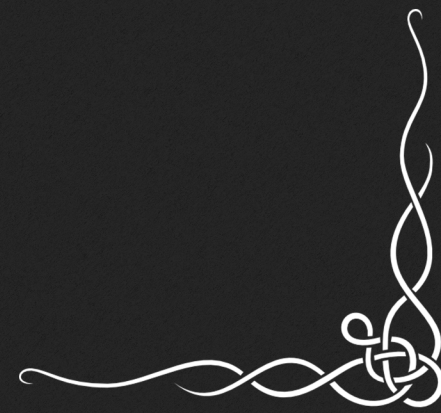
Microsoft Teams, Discord



Logos de los programas utilizados



3. Preproducción



3.1 Conceptualización

El concepto de Valico surge en el inicio de la asignatura de Videojuegos, cuando los integrantes del grupo decidieron formar equipo. Se tenía en mente diseñar un videojuego heredero del espíritu de The Legend of Zelda en 3D y también existía interés en hacer un proyecto que representase la cultura y mitología celta.

Después de investigar el mercado indie, la práctica inexistencia de videojuegos con estas características dieron pie a la conceptualización de Valico: The Grim World; y es que los juegos basados en The Legend of Zelda, resultantes de la investigación, se inspiraban en el estilo cenital clásico de la saga y la temática celta apenas aparecía reflejada en casi ningún título del mercado.

El equipo no recuerda exactamente cómo se llegó al nombre “Valico”, pero claramente surge de la alteración de la palabra “Galaico” que define la temática del videojuego.

La trama fue planteada de forma que cualquier persona pueda conectar con Cassidy. No se quiso crear al típico “héroe elegido por el destino” que lucha por el bien de todos, en cambio se pensó un personaje más normal que lucha por intereses personales, como buscar la cura para la enfermedad de su mejor amigo.

La clave de la historia de Cassidy es que miles de mortales ya pasaron por el mismo camino y fallaron. La Diosa Morrigan elige un mortal que tenga alguna posibilidad contra Bandua y lo guía por un arduo camino, si el mortal muere, simplemente busca uno nuevo. Lo único que hace diferente a Cassidy del resto es que logra derrotar a Bandua.

Para elegir los dioses y criaturas de la historia, el equipo leyó sobre la mitología celta y escogió los seres más interesantes. A partir de ese momento, se pensaron backstories para cada personaje y se empezaron a hilar las ideas.

3.2 Ambientación

El universo del juego está ambientado en un mundo ficcional llamado Valico, que coge inspiración de pueblos reales de Galicia.

En el plano terrenal de Valico hay siete localizaciones a las que el jugador puede acceder: Tarrío, Pazos, A Escravitude, Estramundi, Iria Flavia, Souto do Monte y Cruceiro do Alba. También, los caminos y bosques que conectan cada lugar son zonas explorables.

En el plano divino existen cuatro templos a los que se puede acceder: Templo de Belenus, Templo de Navia, Templo de Dagda y Templo de Lugh.

Para la demo se desarrolló el pueblo de Pazos, lugar ambientado en el principio de la Edad Media o una Edad Antigua muy tardía. Para su creación, se observó cómo funcionaba la urbanística de los yacimientos castrenses en Galicia y de las ciudades medievales europeas.

A partir de estas ideas, se construyó un plano urbano que estratificaba sus edificios y localizaciones en base

a diferentes alturas, imitando los modelos castrenses y con una fuerte adaptación al período medieval: mercado en el centro del pueblo, altas murallas con pasarela y torreones o la inclusión de una fortificación como último método de defensa.



Castro de Viladonga



Castro de Viladonga

Para facilitar la asociación de este universo con Galicia se incluyeron elementos típicos de su cultura tales como un molino de río, distintos cruceiros o la localización del cementerio cerca de un grupo de viviendas, dada la relación atípica del pueblo gallego con la muerte. Como Valico es un mundo ficticio, el equipo tuvo en consideración añadir elementos que pueden resultar anacrónicos, para facilitar la asociación del juego a la cultura galaica.

*Cruceiro*

Además de Pazos, se creó el Bosque a las afueras del pueblo, que conecta con Tarrío y también se hizo la sala final del Templo de Belenus, en donde se desarrolla la batalla contra el jefe.

*Bosque en Castro de Rei, Lugo*

3.3 Mecánicas planteadas

En esta fase se plantearon las mecánicas a desarrollar, para el funcionamiento del videojuego.

Cassidy

A nivel de movilidad se planteó que el personaje debía poder andar, trotar, correr, saltar, ir en cuclillas y trepar muros de poca altura. Para el combate, Cassidy debía ser capaz de fijar objetivos, pelear cuerpo a cuerpo, con espada y con magia elemental, por otro lado en defensa podría bloquear, rechazar golpes o *parry* y rodar.

Además, Cassidy podría utilizar consumibles como pociones, hablar con NPCs e interactuar con elementos como puertas.

Weizo

Para este personaje se plantearon dos mecánicas principales: acompañar al jugador y reproducir música según la situación del juego. Con esto se buscaba tener un acompañante que recordase al

jugador sus objetivos y además introdujese la música de forma diegética.

Enemigos

Los enemigos se plantearon de forma básica, siendo su principal objetivo perseguir al jugador al detectarlo visualmente para acabar con él. Para que el combate fuese algo más dinámico, los enemigos también podrían bloquear o esquivar los ataques.

Dependiendo del tipo de enemigo su comportamiento sería diferente. Para la demo se planteó el desarrollo de Joint-Eaters; grupo de enemigos débiles pero con un comportamiento muy agresivo, y del jefe final, Belenus; enemigo muy fuerte con un estilo de combate mucho más inteligente y paciente.

A mayores, se planteó la posibilidad de desarrollar Esqueletos; enemigos de fuerza media con un estilo de combate dinámico similar al del jugador.

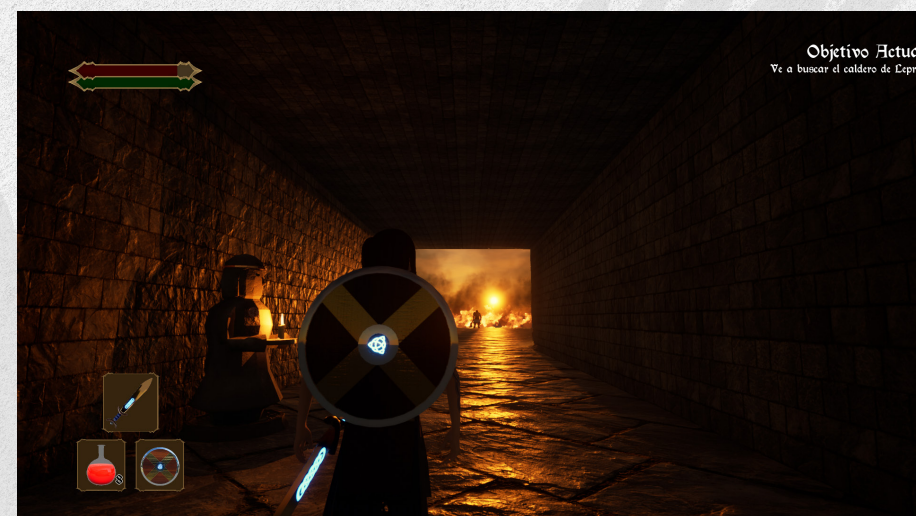
3.4 Creación de la demo

La demo se desarrolló teniendo en mente el futuro juego completo. Con este fin, se creó un entorno como es Pazos, uno de los principales pueblos de Valico, para poder explorar las mecánicas y cómo se interactuaba con el entorno y con la IA acompañante. De esta forma, los objetivos iniciales dentro del pueblo son seguir las indicaciones de Weizo para ir a diferentes lugares, recoger algún objeto o llegar hasta un punto concreto del pueblo.



Mientras que en el segundo nivel, el del Bosque, el objetivo es recuperar el caldero de Lepre, pero el mapa se aborda de manera libre: el jugador puede explorar a su gusto, moverse por el mapa, esquivar o pelear con enemigos. Todo según el ritmo que se busque y la experiencia que más se adapte a cada jugador.

Una vez conseguido el caldero y superado el segundo nivel, el jugador se enfrenta al último reto de la demo: Belenus, el primer jefe de Bandua. (para más información sobre Bandua, consultar el Anexo 5).

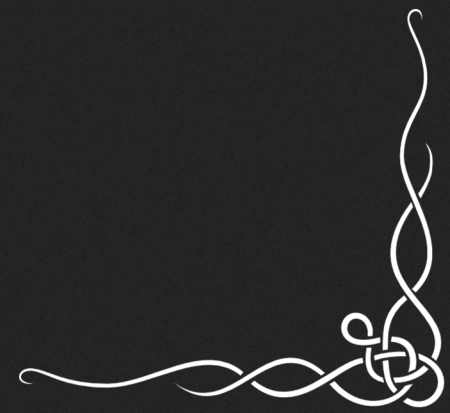
Con esto, el jugador puede disfrutar una experiencia concentrada de lo que sería el juego completo.



Batalla final de la demo



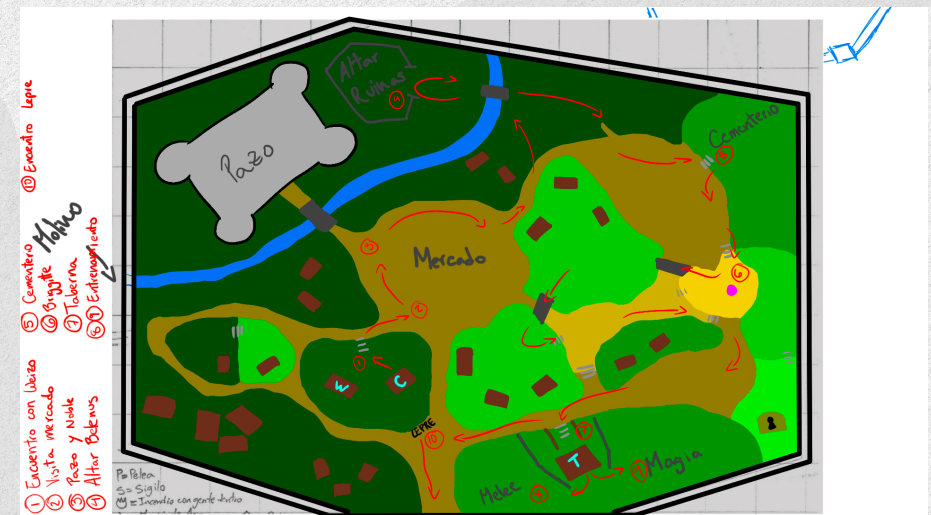
4. Producción



La construcción de los escenarios que se desarrollaron para la demo fue una parte central del proceso de producción.

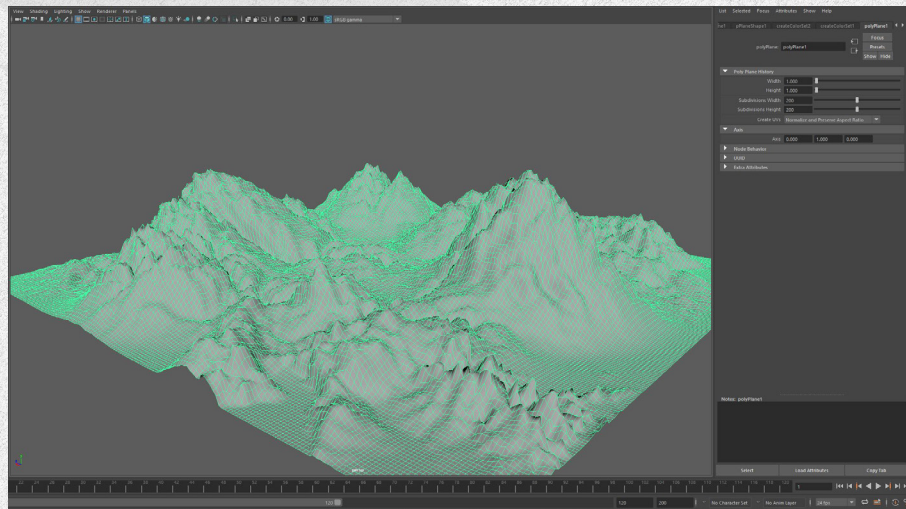
Luego, en Unreal Engine 4, se colocó la imagen de los mapas como referencia para hacer el blocking del nivel. Se utilizaron cubos como placeholders para los edificios y se generó el terreno con la herramienta landscape, para crear las zonas navegables y parte del entorno visible.

Las opciones para deformar el terreno que ofrece esta herramienta son algo rudas, pero para dar las formas más suaves de colinas erosionadas dentro de la ciudad y en parte del entorno, funciona de manera perfecta.



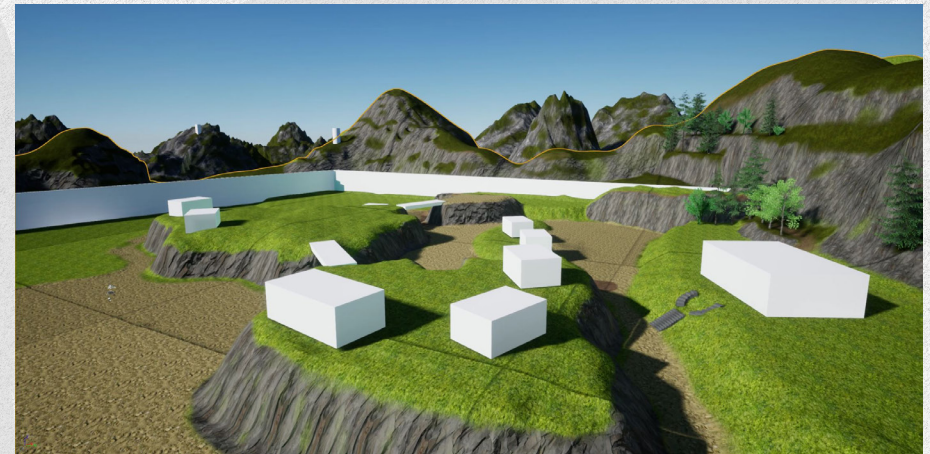
Early concept del Mapa de Pazos.
Para ver el mapa completo consultar Anexo 4

Para generar el terreno no navegable, pero visible en el fondo de los niveles, se decidió utilizar Autodesk Maya. Aquí se generó un planar y con la herramienta de sculpt, usando heightmaps de montañas, se esculpieron picos más escarpados y duros.



Heightmap sculpting

De vuelta en el motor de videojuegos, el último paso fue colocar los materiales. Así, se usó un landscape material para poder pintar texturas según se necesitara. Por otra parte, para las mallas de terreno provenientes de Autodesk Maya, se utilizó la técnica llamada vertex painting para conseguir un acabado similar al del terreno de Unreal Engine 4.



Colocación de materiales al terreno de Pazos

La siguiente parte fue la creación de los bosques y la colocación de los árboles tanto en las zonas jugables; para que diese una ambientación más medieval y natural a la ciudad, como en las zonas exteriores; para simular ese mundo fantástico galaico.

Tras esto, se empezó a sustituir el blocking por mallas detalladas de casas y edificios, para poder tener una visual más acertada de cómo iba quedar el principal nivel jugable.



Emplazamiento de vallas y árboles

Al mismo tiempo, se colocaron vallas en algunas zonas del pueblo para delimitar el espacio de casas o zonas comunes y dar algo más de vida al pueblo.

Tras este paso se modelaron los edificios más importantes como son la casa de Cassidy, la taberna o el pazo y también se empezaron a detallar los caminos con diferentes usos de texturas, alturas, colocación de hierbajos a los laterales o con delimitaciones creadas con árboles para algunos de los caminos.



Colocación de la casa de Cassidy

Las últimas etapas del desarrollo del escenario de Pazos fueron: la creación del río mediante splines, la colocación del molino, la sustitución del blocking de la muralla por mallas específicas, la creación del antiguo lugar religioso y el cementerio de Pazos. Con esto, lo último que quedaba era ambientar con detalles en algunos lugares o dar vida con personajes a los distintas zonas del pueblo.



Creación del río mediante splines

La creación y desarrollo del mapa del Bosque fue considerablemente más sencilla. Se comenzó de la misma forma, con el blocking. La geometría básica funcionó como placeholder de edificios y demás estructuras y se desarrolló el terreno dando alturas a los caminos, colocando árboles y creando un lago utilizando splines, tal como con el río de Pazos. Finalmente se sustituyeron los placeholders por las mallas específicas.

La ruta está claramente marcada por los caminos, por lo que para aumentar la dificultad de la misión, se crearon distintos senderos y se emplazaron varios enemigos a lo largo del mapa.

4.2 Creación de personajes

Para el proyecto se crearon un total de cinco personajes: Cassidy, Morrigan, Weizo, Lepre y Belenus. Un personaje extra que se añadió en una fase más avanzada del proyecto fue Chaun, hermano gemelo de Lepre. Para crearlo simplemente se modificó su vestimenta. También se crearon diferentes personajes genéricos con algunas iteraciones: pueblerinos y guardias para ambientar Pazos y esqueletos para hacer de enemigos.

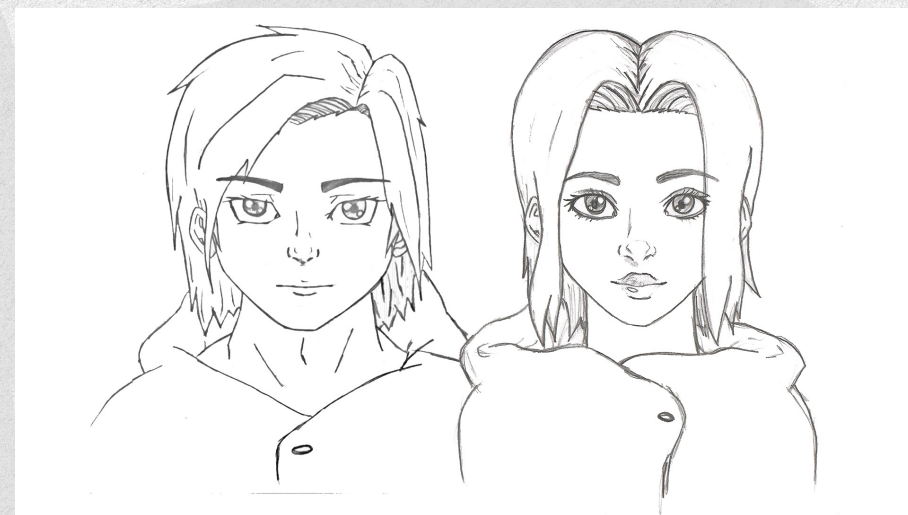
Para diseñar a los personajes, el equipo tuvo en cuenta diferentes aspectos previamente definidos: su rol, personalidad, backstory y el contexto de Valico. Al mismo tiempo, se buscaron referencias (principalmente dentro de la cultura celta) para definir mejor cada idea.

Para ilustrar el resto del pipeline utilizaremos al protagonista de Valico: The Grim World, Cassidy, como ejemplo.

Como el objetivo del equipo es que los jugadores puedan empatizar fácilmente con Cassidy, se

diseñaron dos versiones, una masculina y otra femenina.

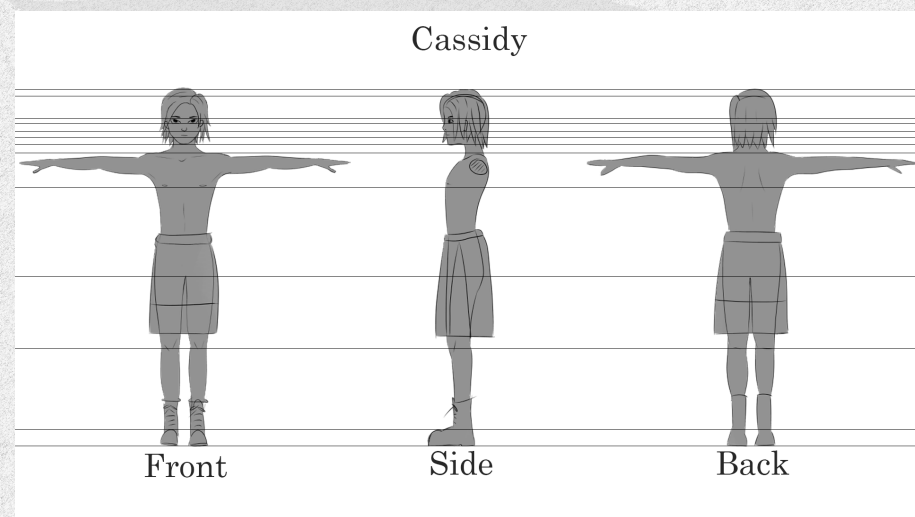
De esta forma, se ofrece la posibilidad de encarnar al protagonista de un modo más flexible y a libre elección del jugador. Dicho esto, para la demo solo se creó la versión masculina.



Concept Art de Cassidy

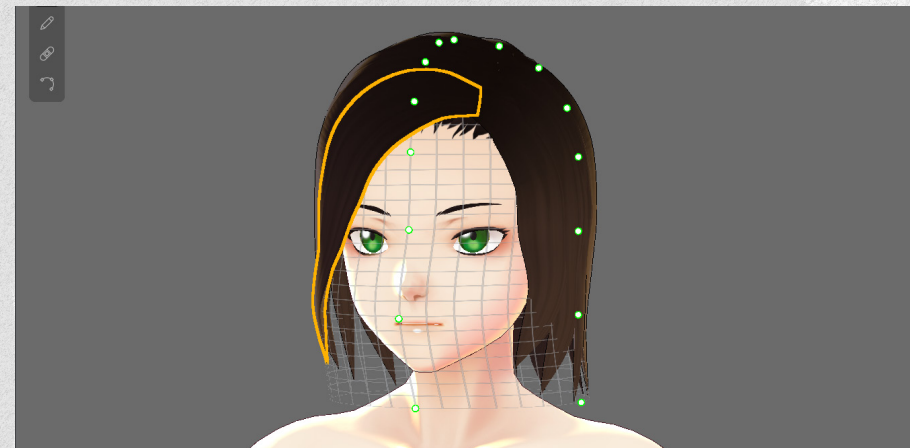
Para su diseño se tomaron referencias de guerreros celtas y magos de combate principalmente.

Una vez terminado el diseño del personaje, se procedió a crear el model sheet que serviría como guía en la creación del modelo 3D.



Model sheet de Cassidy

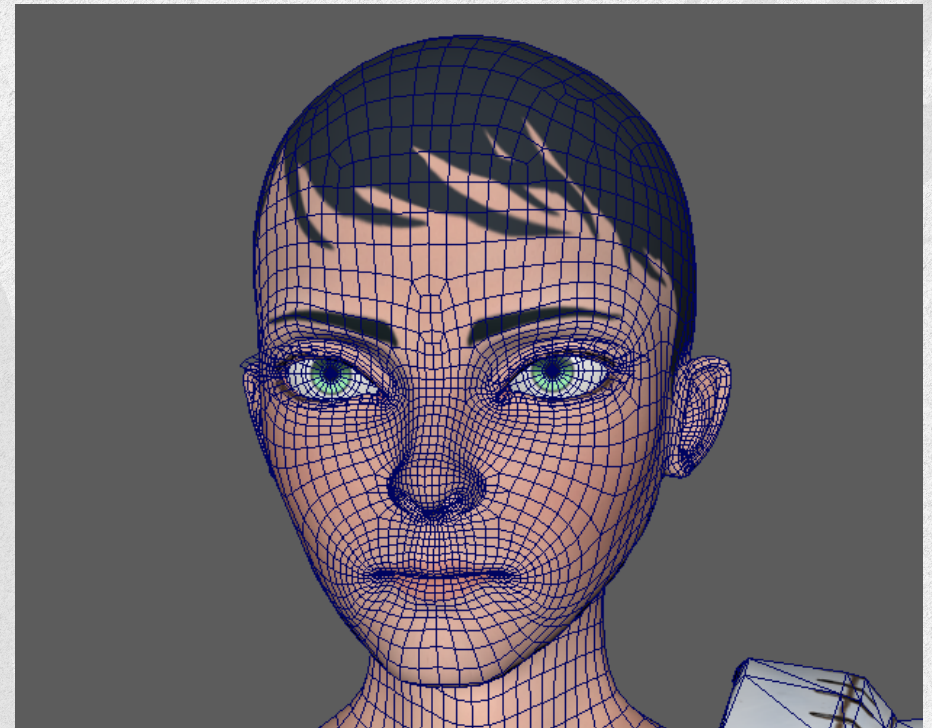
En un primer momento, se utilizó el creador de personajes VRoidStudio para darle vida a Cassidy. Este software resultaba realmente atractivo porque los modelos y texturas estilizadas gustaban para el proyecto, además de la cómoda herramienta de creación de pelo poligonal. Sin embargo, por limitaciones del software y cambios en la estética del juego, el equipo decidió utilizar Adobe Fuse para crear el cuerpo base de Cassidy.



Creación del pelo en VRoidStudio

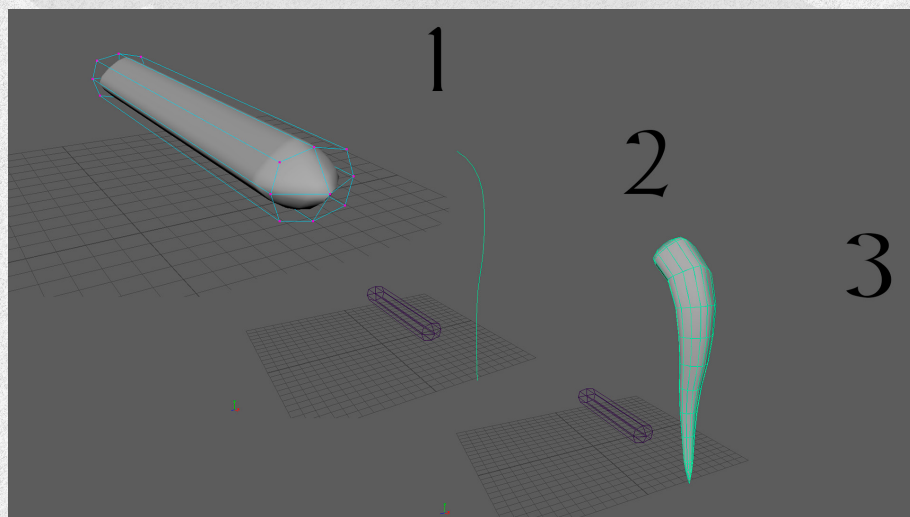


Cassidy en Adobe Fuse



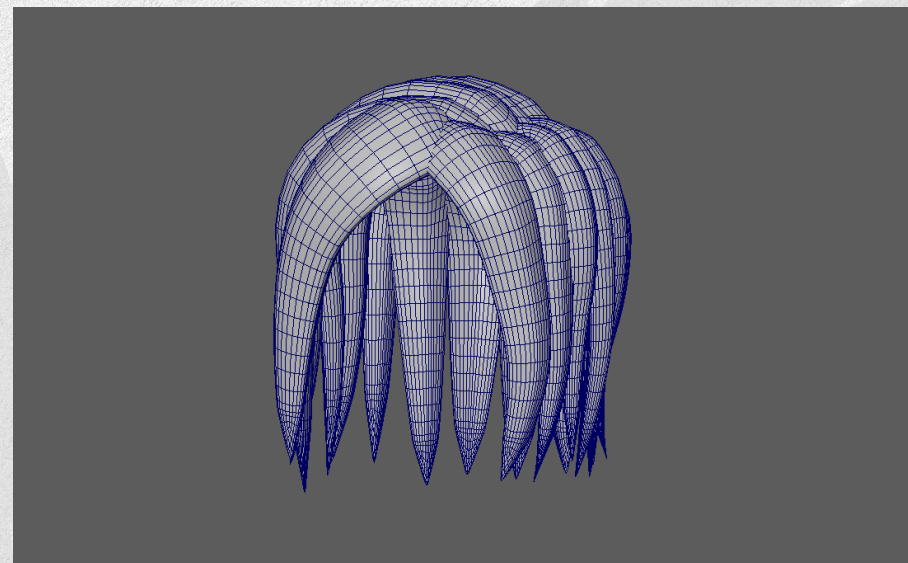
Cassidy en Autodesk Maya

El modelo obtenido se exportó a Autodesk Maya, en donde se modeló el pelo del personaje. Para crearlo se utilizó una técnica bastante cómoda: se creó un polígono base con forma de mechón de pelo, luego se crearon varios “caminos” en la cabeza de Cassidy con CV curves y finalmente se deformó el polígono base para que siguiera las curvas trazadas haciendo uso de la herramienta curve warp.



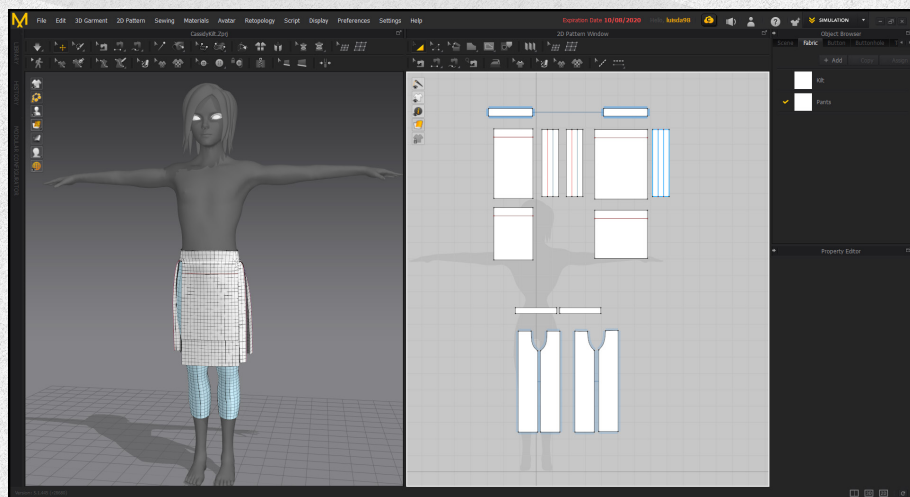
Modelado de pelo con CV Curves

Cuando el polígono sigue el camino definido por las curvas, lo que queda es ajustar la escala y rotación del curve warp y la posición de los control vertex de la curva, hasta obtener el resultado deseado.



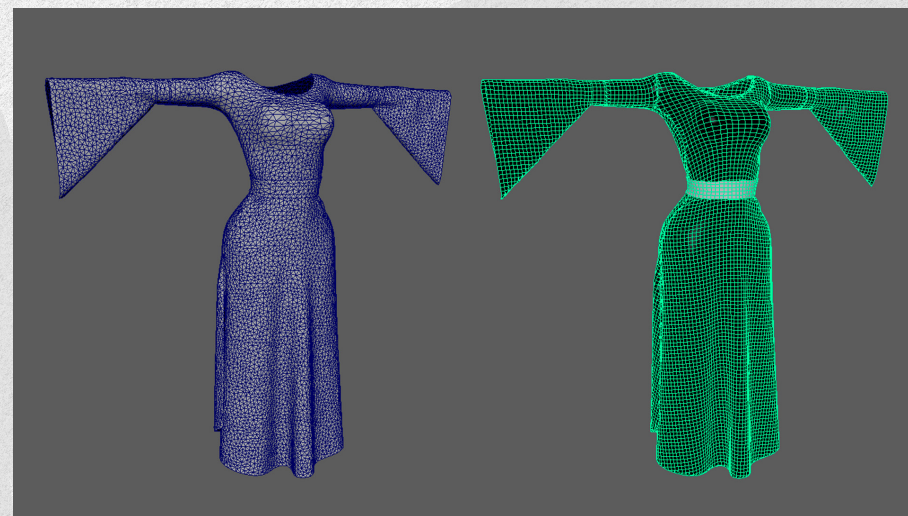
Pelo de Cassidy completo

Para crear la ropa del protagonista se utilizó Marvelous Designer 9. Con este software la producción de la ropa fue realmente cómoda e intuitiva y los resultados obtenidos fueron totalmente satisfactorios. Haciendo uso de la simulación, las piezas de tela toman su forma ajustándose al modelo, dando así grandes resultados.



Vestuario de Cassidy en Marvelous Designer 9

Al terminar la simulación se aplica remesh a la malla, para que la topología sea mucho más ordenada y sencilla. Finalmente se crean los UVs dentro del mismo Marvelous y se exporta el modelo para juntar las piezas en Maya.

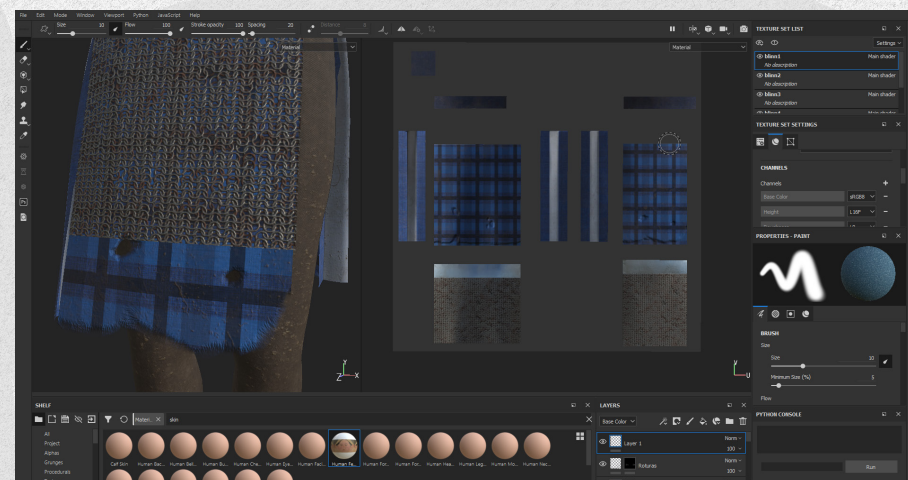


Remesh versión 1 del vestido de Morrigan

Con la ropa importada y el personaje listo, se crean y organizan las UVs restantes y se exporta el personaje completo para hacer el rig con la ayuda del auto-rigger de Mixamo. El gran problema fue la existencia de un kilt. La falda confunde al auto-rigger y el resultado es realmente malo. La solución a la que se llegó, fue hacer el rig a Cassidy sin el kilt en Mixamo, descargarlo, colocarle el kilt y hacer manualmente el skinning de la pieza faltante.

Luego, el personaje es enviado a Substance Painter para realizar todo el proceso de texturizado. Con sus diferentes opciones y herramientas, este proceso fue realmente agradable y satisfactorio. Ensuciar la vestimenta y darle sensación de desgaste fue una de las claves para obtener un buen resultado, sin embargo, la parte más complicada fue texturizar la piel de los personajes. Pese a disponer de los materiales de piel del programa, darle naturalidad a cada zona del cuerpo fue realmente complicado y el equipo considera que es uno de los aspectos que con algo más de experiencia en el campo, en un futuro próximo se puede mejorar.

Para que el personaje cobre vida se utilizaron animaciones de Mixamo y del asset Dynamic Combat System, disponible en la Epic Store. Y finalmente, se pasó el personaje por Blender, en donde se utilizó un plugin que prepara y renombra los joints para funcionar sin errores dentro de Unreal. Básicamente, fue un proceso de limpieza del rig.



Texturizado del kilt de Cassidy

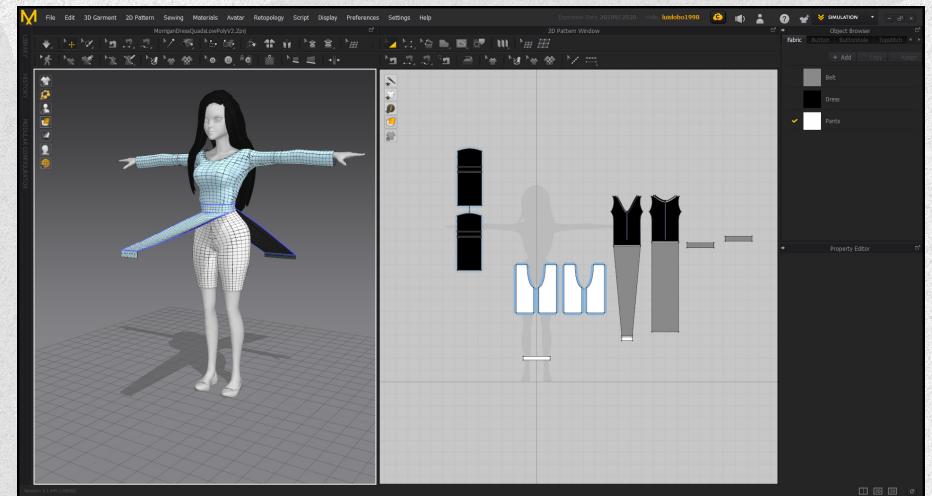
En el resto de personajes se encontraron algunos inconvenientes.

El problema de la falda para hacer el rig fue mucho peor con la creación de Morrigan, pues su diseño original llevaba un vestido largo. Para evitar este problema, agilizar el trabajo y obtener mejores resultados, se rediseñó su vestimenta.



Vestido de Morrigan original

También el largo pelo del personaje dio fallos en Mixamo. Se intentó hacer el mismo procedimiento que con el kilt de Cassidy (skinning post auto-rig), pero al exportar el FBX el pelo estaba fuera de sitio. Luego de varios intentos de arreglar esto, se decidió simplemente juntar ambas piezas dentro de Unreal Engine 4.



Rediseño del vestido de Morrigan

El pipeline de Weizo varía bastante, porque pese a tener forma humanoide es un biosbardo y su figura es totalmente diferente a la de un humano, por esto el modelo fue hecho prácticamente desde cero, solo se utilizó como base el torso de un personaje generado en Adobe Fuse.

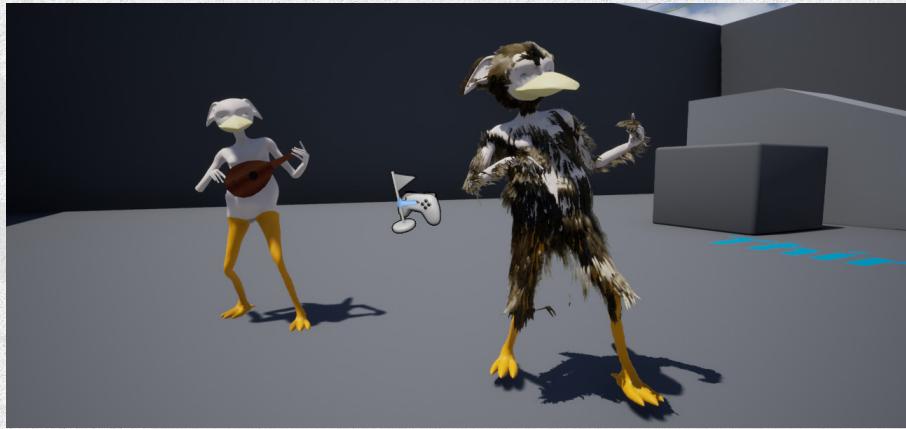
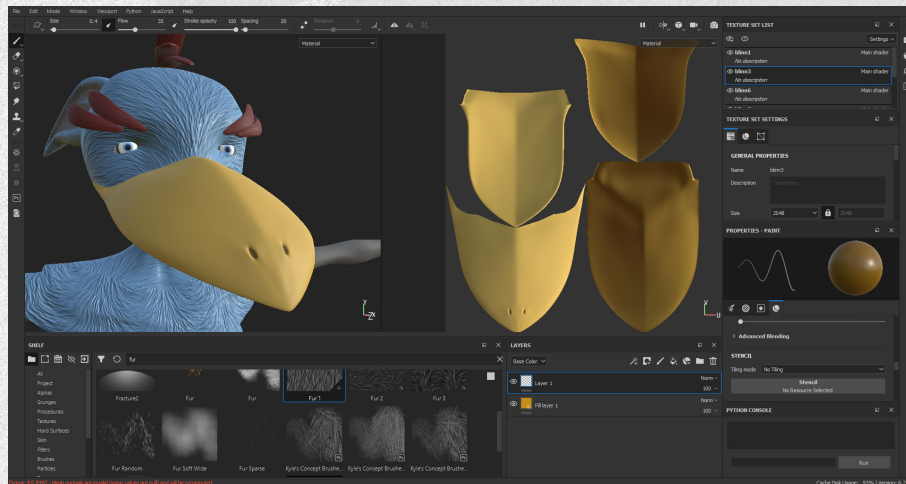


Modelo de Weizo

Otro aspecto a destacar es que este personaje pasó por varias iteraciones hasta llegar al resultado final, porque en un principio la idea era simular las físicas de su pelaje/plumas (creado con XGen en Maya) dentro Unreal Engine 4, utilizando el plugin Allembic Groom Importer, pero el resultado no dejó al equipo totalmente satisfecho y se descartó la idea.



Plumas de Weizo generadas con XGen

*Pruebas de simulación de plumas**Texturizado de Weizo*

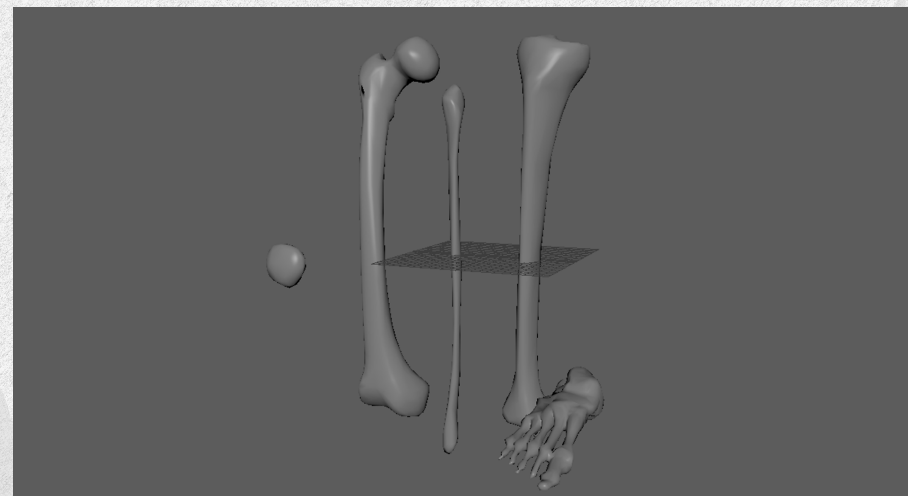
Otra de las ideas que básicamente fue descartada fue la utilización de cloth simulation dentro de Unreal. Se quería simular el movimiento real de la ropa de los personajes al interactuar con las físicas del motor, sin embargo, es un proceso algo inestable y los escasos conocimientos del equipo en ese campo causaron que el experimento solo se llevase a cabo en Morrigan, personaje con una vestimenta sencilla para simular.

*Cloth simulation*

Finalmente, llegó la creación de los enemigos. En la idea original, los enemigos para la demo serían unas criaturas de la mitología celta llamadas Joint-Eaters, pero al no ser humanoides, el proceso de modelado y rigging iba a ser realmente complicado y consumiría demasiado tiempo, así que el equipo decidió crear esqueletos, enemigos algo más fuertes e inteligentes que los Joint-Eaters, pero mucho más fáciles de crear.

Sin embargo, las complicaciones no faltaron, porque modelar un esqueleto realista es bastante más complicado de lo que el equipo pensó al tomar esa decisión. La forma de solventar esta dificultad fue revisar los assets disponibles en las bibliotecas de Epic Games hasta encontrar en uno de ellos piezas sueltas de un esqueleto.

En Autodesk Maya se recolectaron todas las piezas y se armó el esqueleto siguiendo referencias de huesos reales y un model sheet, tal como si de un juguete de LEGO se tratase.



Piezas de la pierna del esqueleto

Una vez finalizado el modelo se intentó llevar a Mixamo para generar un rig base y luego corregir skin-weights en Maya para que cada hueso funcionase como en la realidad, pero la web no comprendió la geometría y no generó el rig.

Así que la solución al problema fue hacer rigging y skinning desde cero usando como referencia el esqueleto de Unreal Engine 4.



Esqueleto con rig

Para el resto de los personajes se sigue básicamente el mismo pipeline que el de Cassidy.

4.3 Creación de props

En primer lugar, el proceso de producción de los props pasó por revisar lo que el equipo tenía disponible en las bibliotecas de Epic Games. Una vez finalizado el proceso, se hizo una lista con los props disponibles y los faltantes.

Luego se acordó cómo serían utilizados cada uno de estos props, ya que se podían usar íntegramente, cambiar las texturas y materiales o modificar también la geometría.

De esta forma, el asset de vegetación se usó tal como vino en el paquete de Epic Games. Por otra parte, el paquete de casas que bañan el pueblo se extrajeron de Unreal, se pasaron por Maya para modificar elementos de su geometría que no casaban con la estética o idea del pueblo (farolas, chimeneas, carteles...) y se modificaron los materiales, UVs y texturas, tanto en Substance Painter como usando materiales extraídos de Quixel Bridge.

Con este sistema se consiguió ahorrar mucho tiempo de la fase de producción, utilizando modelados y props que servían, pero dándoles un toque diferente para que casasen más con la idea global del proyecto.



Casa del asset Medieval Kingdom modificada

Para los props de creación propia se utilizó Autodesk Maya para modelar y crear los UVs y Substance Painter para texturizar.

La espada de Cassidy se diseñó tomando como referencia espadas y dagas de la cultura celta. Se creó un model sheet y a partir de él se modeló el objeto en Maya, para posteriormente texturizarlo en Substance Painter, en donde se agregaron elementos mágicos al arma para diferenciarla de los objetos comunes del juego.

Con el resto de props creados; como el martillo de Belenus, el laúd de Weizo, escudos, etc., se siguió el mismo método que para la espada de Cassidy.

Por otra parte, para algunos props como las lápidas del cementerio o las estatuas de Brighid se crearon varias iteraciones de los materiales, en donde se ensuciaron y marcaron de diferentes maneras, para dar diversidad a objetos que se repetirían varias veces por el mapa.



Espada de Cassidy

4.4 Diseño sonoro

Música

Para elegir el estilo de la banda sonora se escucharon diferentes temas celtas y medievales. Se consideró la adquisición de paquetes de música disponibles en el Marketplace de Epic Games, sin embargo, lo que ofrecían no se ajustaba totalmente a las necesidades del equipo.

Por esto se habló con el compositor, Fernando Rouco, para acordar la creación de la banda sonora. Al compositor se le facilitó un documento con el lore del juego (Anexo 5), para explicar la historia que se quiere contar a través de la música. Con este documento, el equipo se aseguró que Fernando estuviese en pleno conocimiento de la historia del proyecto, haciéndolo, de esta forma, un miembro más del equipo.

En un principio se le pidió una única canción, en la que Weizo en la taberna contase la historia de Belenus y Belisama, además de una versión sin voz para ambientar el lugar.

Posteriormente se acordó la creación de dos arreglos del tema, uno para utilizarse en exploración y combates y otro para zonas de misterio, y otro tema dedicado a la batalla final contra Belenus.

Finalmente, el compositor entregó al equipo un total de seis canciones:

- La leyenda de Belenus y Belisama (3 versiones)
 - Versión con laúd y voz (estilo A)
 - Versión con laúd y voz (estilo B)
 - Versión instrumental
- Tema de exploración y combate (arreglo)
- Tema de misterio (arreglo)
- The Fair Shining One (Batalla contra Belenus)

Efectos sonoros

Una parte importante de la sensación de juego es tener un diseño sonoro que refleje las acciones del jugador.

En la ciudad de Pazos hay un sonido ambiente formado por pistas de audio de pájaros, árboles y viento, para dar la sensación de naturaleza.

Otro efecto sonoro crucial son las pisadas de Cassidy, que cambia dependiendo del terreno que tiene bajo sus pies. Para este proceso, se descargaron pistas de audio libres de derechos de autor, para luego editarlas en Sony Vegas Pro 15.

Se hicieron los cortes necesarios y se panoramizó en un 20% el audio a la izquierda o derecha dependiendo del pie al que corresponde cada sonido. Después de este proceso, el audio se llevó a Unreal Engine 4 para comenzar con el proceso de programación.

En todas las animaciones donde Cassidy da pasos, se colocaron señales para poder referenciarlas en el Animation Blueprint del personaje.

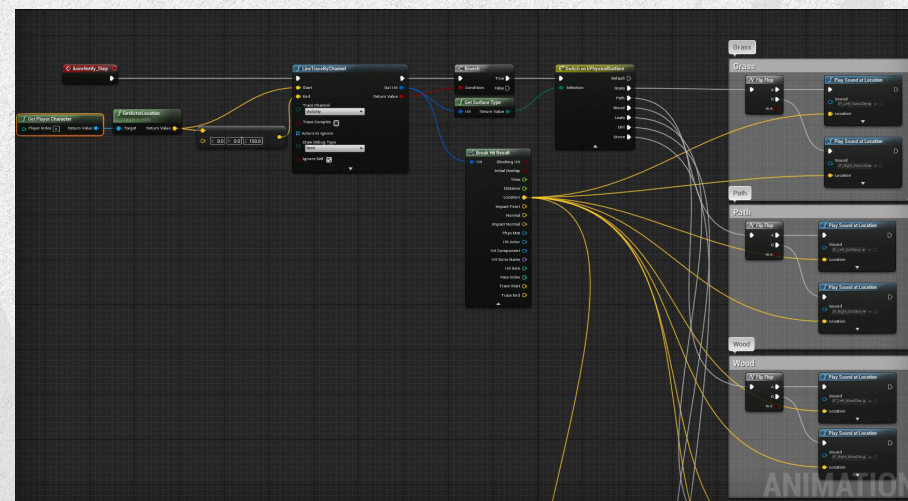
Una vez colocadas todas las señales, se accedió al Event Graph del Animation Blueprint, en donde se creó un Event llamado AnimNotify para que cada vez que una animación llegase al frame donde estaba la señal, se envíe un aviso.

Luego se agregó un nodo Line Trace By Channel con referencia al Player Character. De aquí se pasó a un Branch y se rompió uno de sus nodos para poder sacar la información necesaria (que en este caso era el Location) por separado.

Se creó un nodo Get Surface Type que se conectó a un Switch on EPhysicalSurface que tenía la información de todos los Physical Materials que se crearon, después de esto lo único que faltaba era añadir un Flip Flop para que variase los pasos entre izquierda y derecha y hacer un Play Sound at Location dándole la Location del Line Trace By Channel.

Este proceso fue posible gracias a los Physics Materials que tiene el motor, que permite darle una referencia física a cada material deseado, para poder variar los sonidos cuando un actor pisa encima.

Todos los sonidos tienen derechos de libre uso y pese a que algunos para la demo se han dejado como placeholders, para centrarse más en otros aspectos, no quita que el proceso de creación e implementación fuese importante.



Blueprint de comprobación de Physics Materials

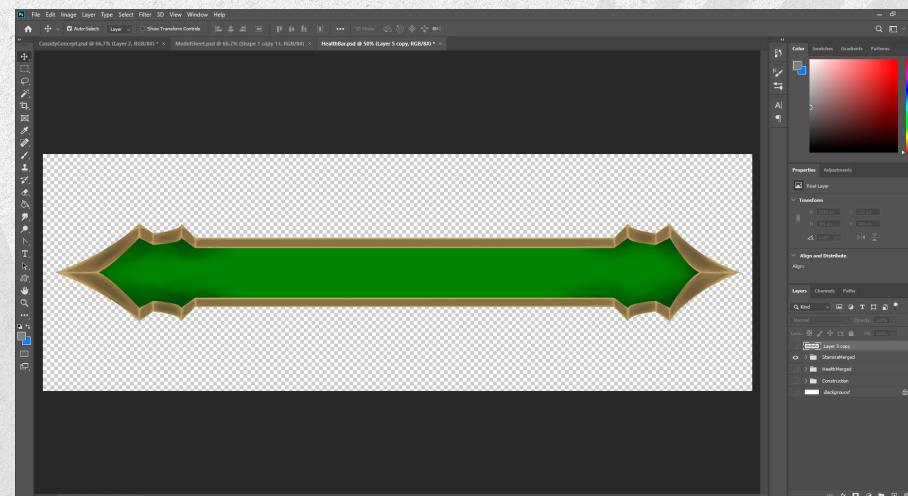
4.5 Interfaz

La interfaz fue diseñada tomando como referencia juegos como The Legend of Zelda, The Witcher y Dark Souls. Se dibujaron diferentes bocetos y, posteriormente, dentro del motor el equipo procedió a realizar diferentes pruebas, incluso durante el desarrollo de los proyectos para la asignatura Interacción 3D.

En una etapa más avanzada del desarrollo, las interfaces básicas fueron sustituidas por las que el asset Dynamic Combat System ofrecía. Se tomó esta decisión porque el funcionamiento era muy similar y además bastante cómodo de manejar.

Las texturas para la interfaz fueron creadas en Adobe Photoshop. Allí se dibujaron y pintaron para dar una estética rústica y medieval que encajase con las necesidades del proyecto.

Finalmente, en Unreal se sustituyeron los gráficos de Dynamic Combat System por los propios.



Creación de las barras de energía y salud en Photoshop

4.6 Jugabilidad y mecánicas

Valico: The Grim World es un videojuego en tercera persona, en donde la orientación del personaje y la cámara son independientes y controlables por el jugador, es decir, si el personaje gira hacia la izquierda de la cámara, ésta no rotará si el jugador no decide girarla.

Exploración

En este modo de juego, el jugador es libre de moverse por todo el mapa en donde podrá conversar con los distintos NPCs, buscar tesoros, interactuar con objetos, etc. El objetivo principal es cumplir misiones, que consisten en visitar diferentes zonas de Pazos. Toda la creación de eventos se hizo dentro del level blueprint, haciendo que cada uno ocurra en un orden específico.

Para las interacciones con el entorno, se construyeron diferentes blueprints que entran en funcionamiento cuando el jugador presiona el input específico dentro de un trigger volume. Con este método se crearon todos los actores interactivables.

Para el sistema de diálogos el equipo adquirió un asset disponible en la tienda de Epic Games, llamado Extreme Dialogue System, que se activa del mismo modo que el resto de actores (Input press dentro de un trigger volume). Se decidió utilizar este asset, ya que trae de base unos blueprints muy sencillos, que utilizan datatables para cambiar idiomas (que en un futuro resultaría de gran utilidad) y diálogos avanzados que permiten dar opciones de respuesta al jugador o colocar cámaras para cambiar de plano durante los diálogos.



Exploración de Pazos

Trabajar con este asset fue realmente cómodo para llevar la narrativa de la demo, además de que se ahorró bastante tiempo de desarrollo.

Combate

En este modo, Cassidy puede atacar a los enemigos haciendo uso de sus puños y espadas, con el escudo puede aguantar los golpes hasta que se agote la barra de energía o intentar realizar un parry para desestabilizar al rival, también puede evadir los ataques rodando, saltando y corriendo.

El combate dentro del juego funciona haciendo uso del asset mencionado anteriormente, Dynamic Combat System (DCS). Sin embargo, para llegar a este resultado el equipo realizó diferentes pruebas.

En un principio la idea era utilizar las animaciones del asset gratuito Advanced Locomotion System (ALS) como base, para dar mayor naturalidad al movimiento de Cassidy y posteriormente agregar animaciones de combate de Mixamo, sin embargo,

no se ajustaban a las exigencias del equipo. Luego de sopesar varias opciones, se llegó a la conclusión de adquirir Dynamic Combat System.

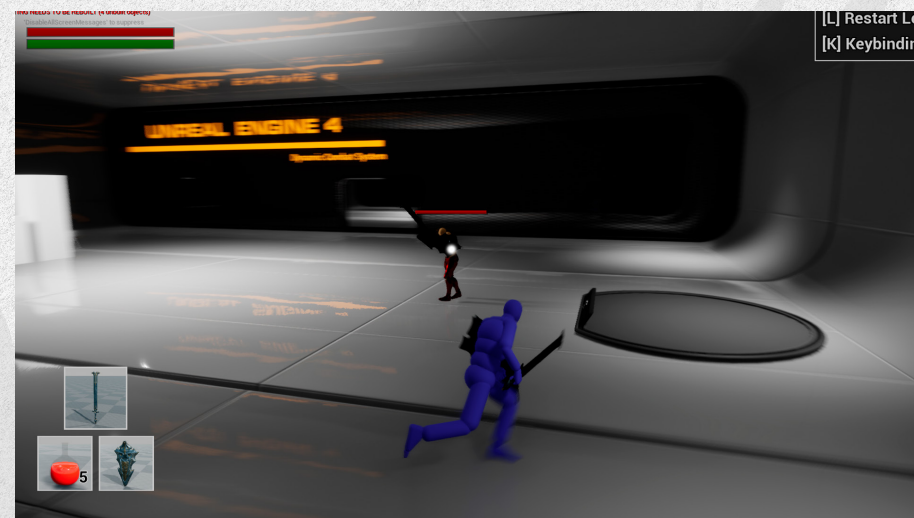
Éste, además de tener buenas animaciones de combate con espada, ofrecía un sistema que se ajustaba a lo que el equipo quería conseguir.



Combate contra esqueletos

El equipo procedió a fusionar ambos sistemas para conseguir el movimiento fluido del ALS y el combate del DCS. El proceso de fusión de blueprints fue bastante tedioso y complicado, pero finalmente se consiguió. Lo único que quedaba por hacer era sustituir el modelo por defecto del ALS por Cassidy. Se hizo todo el retarget de esqueleto necesario, pero el sistema fusionado no entendía correctamente la sustitución de esqueletos y dio muchos errores, como por ejemplo hacer todas las animaciones del ALS pero no ser capaz de reproducir las del DCS.

Luego de mucho tiempo intentando hacer que el sistema funcionase con Cassidy, el equipo decidió que, por cuestión de tiempo y para poder avanzar en el resto de elementos del juego, ALS se dejaría de lado y se utilizaría únicamente el DCS. De esta forma se perdieron algunas mecánicas como trepar e ir en cuclillas.



Advanced Locomotion System + Dynamic Combat System

La intención del equipo era desarrollar dos estilos de combate: uno cuerpo a cuerpo y otro con magia. Sin embargo, por falta de tiempo, en gran parte utilizado en intentar que ALS+DCS funcionase, el combate mágico tuvo que ser dejado de lado.

Aunque uno de los miembros del equipo desarrolló un prototipo de combate mágico en la asignatura Interacción 3D, pulirlo supondría utilizar demasiado tiempo del que no se disponía.



Prototipo de sistema de magia

También, pese a que Dynamic Combat System tiene una expansión para desarrollar combates mágicos, su adquisición suponía una inversión que el equipo no se podía permitir.

Finalmente, Cassidy cuenta con cinco tipos de golpe con la espada: ataque básico, ataque con giro, estocada, ataque por la espalda y ataque con salto. En combate desarmado cuenta con tres golpes diferentes: puñetazos básicos, patada y puñetazo en el aire. En defensa cuenta con: bloqueo con escudo, parry, rodar y paso hacia atrás.

Blueprints del nivel

Puertas:

Para crear las puertas del juego, se utilizaron trigger volumes para que cuando el jugador se superponga a ellos, pueda accionar el funcionamiento de la puerta utilizando la tecla E. Para crear la animación de la puerta se utilizó el nodo timeline.

Interiores:

Uno de los problemas de los interiores es el mal funcionamiento de la cámara en estos espacios tan reducidos. Por esto, y para dar más naturalidad al comportamiento de Cassidy en los interiores, se creó un blueprint que acerca la cámara al personaje, obliga al jugador a moverse andando y deshabilita el resto de acciones como golpear, rodar, saltar, etc.

Para su creación se utilizó un trigger volume y variables de tipo boolean para indicar cuándo el jugador entra; y sus inputs deben ser deshabilitados, o sale de la zona marcada; para devolver las acciones al jugador.

Este blueprint también se utilizó en otros puntos específicos de Pazos como el cementerio.

Checkpoints:

Los checkpoints en Valico son las estatuas de Brigid. Cuando el jugador entra en el trigger volume, el archivo de guardado se actualiza con los datos almacenados en el Game Instance. Si Cassidy muere, hará respawn en el último checkpoint alcanzado.

Dicho esto, este blueprint no funciona como punto de guardado real, porque solo se consiguió guardar la localización del Player Character y sus variables (equipo, vida e inventario), pero no los eventos de los niveles ni las variables almacenadas en el Game Instance. Luego de varios intentos de arreglar el sistema, se optó por no hacer un guardado real, pero si tener disponible la mecánica de respawn.

Cambio de mapa:

Para poder cambiar de mapa, el jugador deber acercarse a la entrada del pueblo y presionar la tecla E. Funciona cuando el jugador entra en el trigger volume y activa el evento con un input action. Al activar estos blueprints también se guarda la partida.

Sin embargo, esta función estará bloqueada hasta que el jugador haya cumplido una serie de misiones. Para definir si el jugador superó las pruebas, se crearon variables de tipo boolean. Cada vez que se cumpla una misión la variable pasará a verdadera y cuando todas sean así, el funcionamiento del cambio de mapa estará disponible.

Pickups:

Los pickups son actores con los que el jugador puede interactuar pulsando la tecla V. Estos blueprints almacenan diferentes objetos que Cassidy puede recolectar, utilizar y equipar. El contenido de los pickups es totalmente personalizable según lo que el equipo de desarrollo quiera otorgar al jugador.

El blueprint proviene de Dynamic Combat System y para personalizarlo el equipo siguió las indicaciones de la documentación que el asset provee.

Para crear los objetos del juego, se utilizaron los blueprints base de DCS, que permiten una total personalización tanto a nivel visual como funcional. En estos blueprints se puede definir cuánto daño hace una espada, cuánto soporta un escudo, qué bonificadores dan los objetos equipables, etc.

Activación de cinemáticas:

Las cinemáticas se activan cuando el jugador entra en un trigger volume. Durante la cinemática se desactivan los inputs del jugador, para evitar que pueda moverse mientras la cinemática ocurre.

Para el desarrollo de las cinemáticas menores, se utiliza el Extreme Dialogue System. Su sistema multicámara resulta realmente cómodo para hilar las conversaciones de los personajes y expresar de forma más interesante la historia que se quiere contar.

En el proyecto se trabaja con dos tipos de inteligencia artificial: acompañante y enemigos

Weizo es la IA acompañante. Para su funcionamiento se programó su blueprint utilizando el nodo AI go to, en el que se le indica que siempre debe ir a la localización de Cassidy. Para que Weizo no interfiera con la visión y movimientos del jugador, se estableció una distancia de seguridad aplicando un offset a la localización exacta de Cassidy.

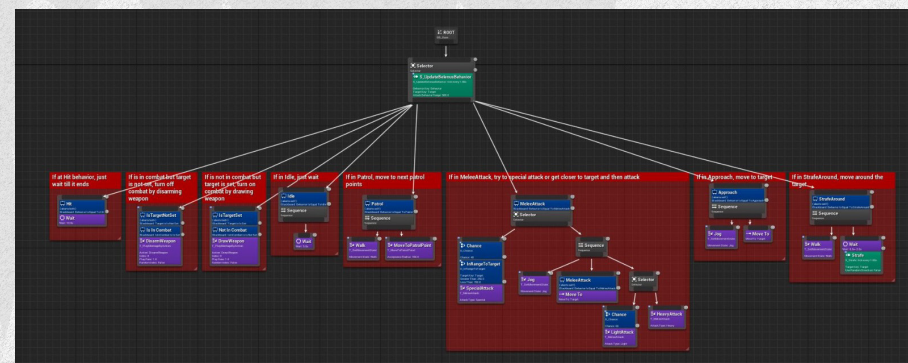
Los enemigos que aparecen en la demo son Esqueletos y el jefe final, Belenus.

Ambos están hechos con los blueprints de Dynamic Combat System, en los que el comportamiento está definido por Behavior Trees o Árboles de Comportamiento.

Lo principal es fijar un objetivo, que en este caso será Cassidy. Una vez fijado lo perseguirá y atacará.

Gracias al Behavior Tree, los enemigos no siguen un mismo patrón de ataques, permitiendo así, que su comportamiento sea bastante aleatorio y orgánico.

Para hacer distinción entre los Esqueletos y Belenus, se alteró un poco el comportamiento de cada uno. Los que llevan escudo son bastante defensivos, los que solo llevan una espada tienen un comportamiento ofensivo y Belenus, al ser un Dios poderoso, es bastante más agresivo y resistente.



Behavior Tree de Belenus

4.7 Cinemáticas

Para la demo se crearon dos cinemáticas principales: la inicial, en donde se resumirá la historia de Valico, y otra para el concierto de Weizo en la taberna.

Cinemática inicial

Para crear esta cinemática el equipo tomó como referencia *The Legend of Zelda: Breath of the Wild*, en particular, la inspiración salió de la cinemática en donde Impa cuenta a Link la historia de Ganon haciendo uso de una pintura.

En la cinemática inicial se cuenta la historia de Valico a través de una bandera/estandarte. Para la creación de ésta, el equipo contó con la colaboración de Sofía González, estudiante de Bellas Artes de la Universidad de Vigo. Al igual que con el compositor, a la artista se le facilitó el documento de lore de Valico.

Simultáneamente se hizo el guion literario de la cinemática.

Una vez finalizado el guion literario y con el diseño de la bandera listo, se procedió a la creación del guion técnico y storyboard.

Con todo listo, el equipo procedió a montar la cinemática utilizando Adobe Premiere. Pero el resultado final no gustó en absoluto y se desechó la cinemática. Sin embargo, la bandera diseñada igualmente forma parte del videojuego.

Se replanteó la cinemática manteniendo la narración planteada en el guion literario y con un nuevo storyboard. La idea a la que se llegó fue contar la historia a través de dibujos. En Photoshop se crearon diferentes capas para cada parte de los dibujos, y así poder animar las piezas por separado y dar dinamismo a la cinemática.

En Premiere se juntaron las piezas, se animaron las capas de los dibujos y se agregó el sonido. Posteriormente el vídeo se agregó en un widget de Unreal Engine 4.



Partes del primer plano de la cinemática inicial

Cinemática del concierto

Para la creación de esta cinemática se tomó como referencia una escena de The Witcher 3: Wild Hunt. En ella, el protagonista, Geralt asiste al concierto de Priscilla. Lo más importante que el equipo sacó de esta referencia fue cómo hacen los estudios profesionales para entretener al jugador durante tanto tiempo, ya que, un concierto es algo que a nivel cinematográfico no da tanto juego.

Para esta cinemática se creó un guion literario teniendo como base la letra de la canción compuesta por Fernando Rouco. Posteriormente, se creó el guion técnico y el storyboard.

Una vez hecho esto, se procedió a crear la cinemática dentro de Unreal Engine 4, haciendo uso de la herramienta sequencer.

Para la actuación de Weizo, se destruyó a la IA acompañante y se creó un actor Weizo nuevo para reproducir las animaciones de la cinemática.

Se hizo esto, porque el comportamiento establecido en el blueprint del Weizo inteligente interfería con el sequencer.

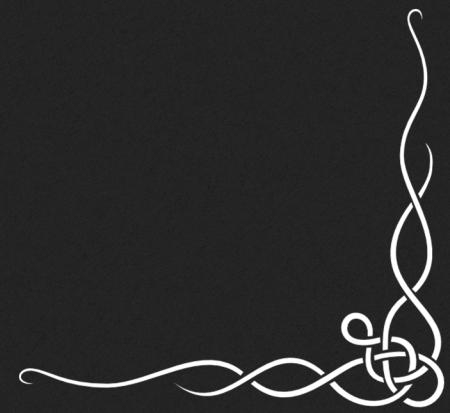
El resto de cinemáticas menores se hicieron sin tanta preparación previa, haciendo uso del Extreme Dialogue System.



Cinemática del concierto



5. Postproducción



5. Postproducción

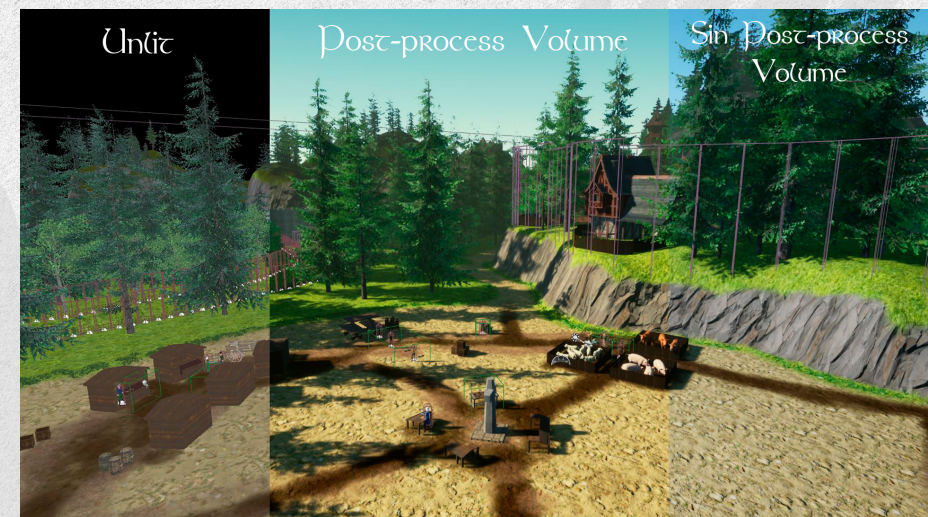
En la etapa de postproducción, se ajustaron todos los valores referentes a la iluminación. Fue realizada con Dynamic lighting (luces móviles), bastante recomendada para mapas amplios y al aire libre. De esta forma se ahorró el tiempo de cálculo y construcción de la luz, a cambio de un poco de exigencia al ordenador.

Para personalizar aún más la iluminación se usaron post-process volumes, que permitieron jugar con volumetric fog, LUT maps, entre otros.

En el mapa de Pazos se buscó dar calidez al lugar, al ser un sitio sin peligro y amigable para explorar.

En el Bosque se quiso dar una imagen fría para que el jugador sienta algo de temor al explorar una zona misteriosa y llena de peligros.

Finalmente en la sala de Belenus, se quiso transmitir la agresividad del fuego relacionada con el Dios y la violencia.



Trabajo de iluminación y color:
Unlit vs Post-process Volume vs Sin Post-process Volume



6. Testing



6. Testing

Durante el desarrollo de la demo, el equipo buscó varias personas para probar el juego periódicamente.

Las pruebas las hicieron sin ayuda o indicaciones del equipo. Como resultado de estas pruebas el equipo encontró algunos bugs, errores de comprensión de la historia y misiones, etc.

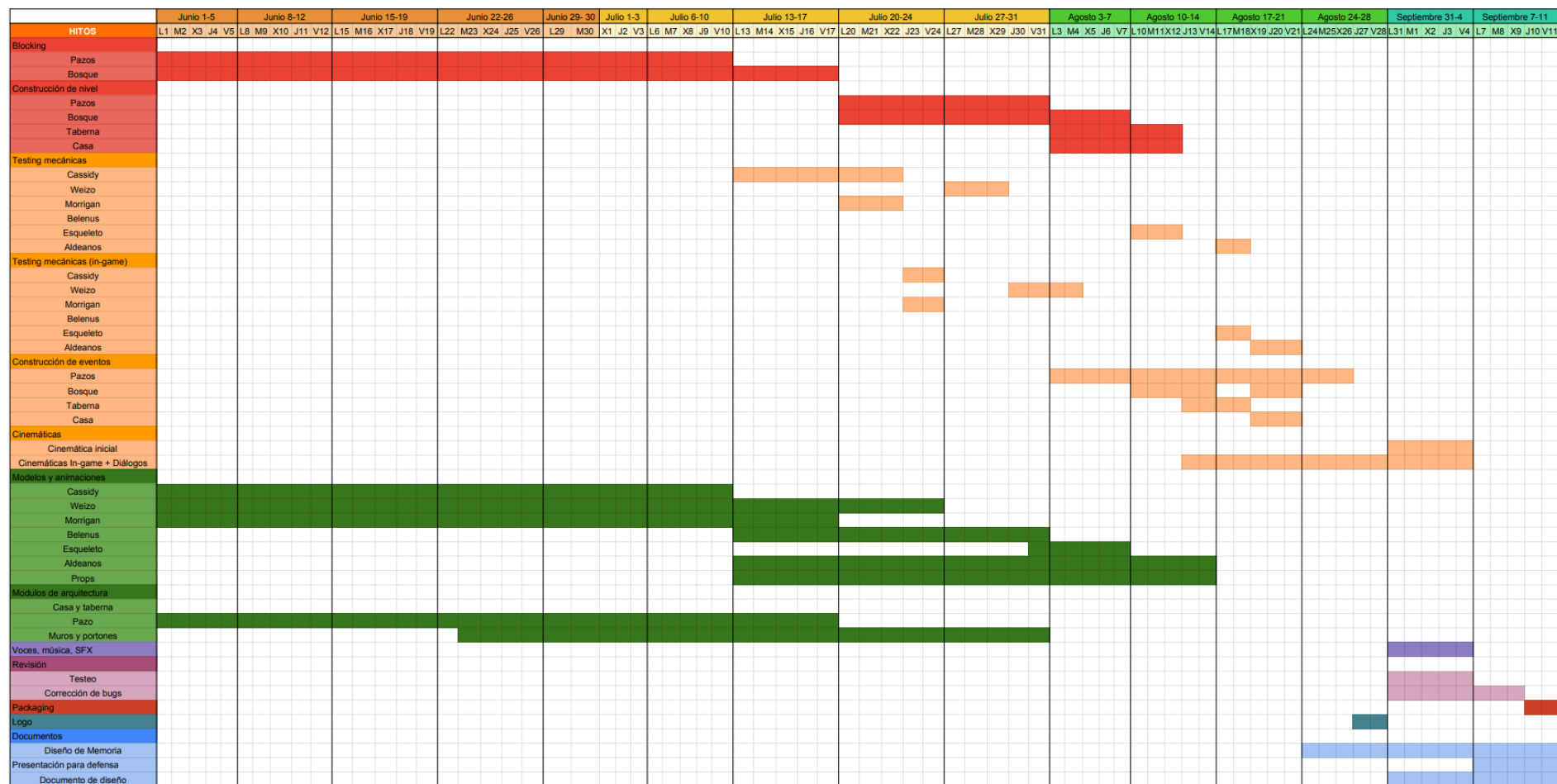
El equipo considera esta fase fundamental para el crecimiento del proyecto, porque el feedback que aportan personas externas es realmente importante. En el desarrollo posterior a la entrega del TFG, se continuarán realizando estos tests periódicamente para asegurar el buen crecimiento del proyecto.



7. Plan de producción



7.1 Calendario de hitos



7.2 Presupuesto

EQUIPO DE PRODUCCIÓN

Personal	Unidad	Nº de Unidades	Precio Unitario	Total
Departamento de Arte				
Concept Artist	Mes	2	1.700 €	3.400 €
Departamento de Diseño				
Artista 3D	Mes	3	1.500 €	4.500 €
Enviroment Artist	Mes	3	1.600 €	4.800 €
Producción				
Modelador	Mes	3	1.100 €	3.300 €
Rigger	Mes	3	3.200 €	9.600 €
Animador	Mes	3	1.400 €	4.200 €
Programador	Mes	4	1.750 €	7.000 €
Gasto Total				36.800 €

Equipamiento	Unidad	Nº de Unidades	Precio Unitario	Total
Hardware				
Pc Sobremesa 1	2.200 € 6 meses		37 €	220 € *
Pc Sobremesa 2	2.200 € 6 meses		37 €	220 € *
Software				
Unreal Engine 4.24	Libre Uso	6 meses	-	0 €
Autodesk Maya 2020	Mensual	3 meses	267 €	801 €
Substance 3D	Mensual	3 meses	20 €	60 €
Marvelous Designer 9	Mensual	2 meses	50 €	100 €
Blender	Libre Uso	3 meses	-	0 €
Mixamo	Libre Uso	3 meses	-	0 €
Adobe Creative Cloud	Mensual	6 meses	20 €	120 €
Google Drive	Libre Uso	6 meses	-	0 €
Office 365	Mensual	6 meses	10 €	60 €
Sony Vegas Pro 15	De por vida	2 Meses	260 €	260 €
VRoidStudio	Libre Uso	3 meses	-	0 €
Gasto Total				1.841 €

*Coste de amortización de este equipamiento considerando una vida útil a 5 años.

Gastos Externos	Unidad	Nº de Unidades	Precio Unitario	Total
Unreal Marketplace				
Dynamic Combat System	Unreal Asset	1	50 €	50 €
Extreme Dialogue System	Unreal Asset	1	10 €	10 €
Banda sonora original				
Horas en el proyecto	Horas	40	7,43 €	297,20 €
Mezcla y Mastering/Nº de canciones	Canción	5	50 €	250 €
Software y mantenimiento	Unidad base	1	50 €	50 €
Gasto Total				657,20 €



8. Conclusiones



8. Conclusiones

Para concluir nos gustaría expresar en primera persona lo que significó hacer este trabajo.

El desarrollo de Valico: The Grim World fue una experiencia inolvidable. Dar vida a una idea tan personal resultó gratificante y al mismo tiempo frustrante, principalmente porque nos consideramos personas bastante perfeccionistas y la existencia de fallos y aspectos mejorables no nos permitieron descansar la mente en todos estos meses de desarrollo.

Las mayores dificultades con las que nos encontramos fueron principalmente por falta de experiencia, además de que siendo solo dos personas, cubrir cada aspecto del trabajo de la mejor manera posible resulta muy difícil.

La situación sanitaria que vivimos (y seguimos viviendo) supuso un gran desafío. Haber estado tanto tiempo encerrados y con la vida “paralizada” trajo consigo muchísimo agotamiento a nivel emocional y mental.

Sin embargo, en este viaje hemos aprendido muchísimo, tanto por tener que resolver problemas inesperados, como por querer alcanzar un mejor nivel al proponer ideas personales y ambiciosas. El aprendizaje es algo que consideramos de vital importancia al realizar cualquier proyecto.

Tristemente, por falta de tiempo y recursos, varias de las ideas originales tuvieron que ser descartadas para la demo, pero en un futuro las desarrollaremos teniendo toda la experiencia obtenida como apoyo.

También, queremos agradecer a Fernando Rouco y Sofía González por apoyar el proyecto haciendo un trabajo fantástico y de altísimo nivel, siempre con un comportamiento profesional.

Además, queremos destacar a la comunidad de Unreal Engine 4. Sin conocernos, demostraron gran interés en ayudarnos sin pedir nada a cambio. Todas estas interacciones nos demostraron que la comunidad de desarrolladores es un lugar en el que podemos trabajar y crecer como profesionales.

Desarrolladores independientes, YouTubers y personas con poca experiencia tuvieron los brazos abiertos para recibir todas nuestras dudas. Sin ellos, Valico no hubiese sido posible.

Gracias a todos, en especial a Jazary, Ricardo, Moritz Naser y Lord Xenu.

Pusimos todo nuestro esfuerzo y corazón en este proyecto, estamos realmente orgullosos del resultado conseguido y esperamos que nuestro juego sirva como referencia para futuros TFG.



9. Bibliografía



9. Bibliografía

Nombres de Dioses y Guerreras Celtas | Mitología Gallega. CD. Consultado en 2020, de <https://celtadigital.com/dioses-celtas/>.

Bandua. Gl.wikipedia.org. Consultado en 2020, de <https://gl.wikipedia.org/wiki/Bandua>.

BELENUS » Dios del sol Celta. Mitologia. Consultado en 2020, de <https://www.mitologia.info/belenus/>.

Belenus. En.wikipedia.org. Consultado en 2020, de <https://en.wikipedia.org/wiki/Belenus>.

BELISAMA o BELISMA. Louespiritucelta.blogspot.com. Consultado en 2020, de <http://louespiritucelta.blogspot.com/2009/12/belisama-o-belisma.html>.

Belisama, Diosa celta ADORADA EN MEDIA EUROPA. Nombres de diosas. Consultado en 2020, de <https://nombresdediosas.com/belisama/>.

Belisama. En.wikipedia.org. Consultado en 2020, de <https://en.wikipedia.org/wiki/Belisama>.

Beltane. En.wikipedia.org. Consultado en 2020, de <https://en.wikipedia.org/wiki/Beltane>.

Descubre todo sobre Belenus, dios mitológico Celta. Conozcamos Todas Las Mitologías De Nuestro Planeta. Consultado en 2020, de <https://hablemosdemitologias.com/c-mitologia-celta/belenus/>.

DevSquad. YouTube. Consultado en 2020, de <https://www.youtube.com/user/VirtusEdu>.

Jazary. YouTube. Consultado en 2020, de <https://www.youtube.com/channel/UCxNqSOPOpvv2k22EN8Xq4xA>.

Jorge, D. M. E. N. P. (2015). La casa medieval en la península ibérica (1.a ed.). Sílex.

Naser, M. Moritz Naser. YouTube. Consultado en 2020, de <https://www.youtube.com/channel/UCYfmdsT0mxMQGyd9wFEO6sA>.

Szewczyk, G. Grzegorz Szewczyk. YouTube. Consultado en 2020, de <https://www.youtube.com/user/morty2632>.

Szewczyk, G. Home Page. Ue4dcs.com. Consultado en 2020, de <https://ue4dcs.com/>.

Santos, J. L. C., Seoane, M. G., & Seone, M. G. (2018). Galicia inmaterial, Galicia máxica. Hércules de Ediciones.

The Dagda. En.wikipedia.org. Consultado en 2020, de https://en.wikipedia.org/wiki/The_Dagda

Two Deities of the Fair Folk, Lugh and the Morrigan. Faerie-world.org. Consultado en 2020, de <http://www.faerie-world.org/tales/deities.html>.

Uwunti. YouTube. Consultado en 2020, de <https://www.youtube.com/channel/UCcmt2iA1hkD9rauy5dAjF8A>.

Wadstein, M. Mathew Wadstein. YouTube. Consultado en 2020, de <https://www.youtube.com/channel/UCOVfF7PfLbRdVEm0hONTrNQ>.